L Number	Hits	Search Text	DB	Time stamp
1	1	("5249783").PN.	USPAT; US-PGPUB	2003/11/26 10:46
2	1	(("5249783").PN.) and high	USPAT; US-PGPUB	2003/11/26 10:46
3	1	("5246783").PN.	USPAT; US-PGPUB	2003/11/26 10:46
4	1	(("5246783").PN.) and high	USPAT; US-PGPUB	2003/11/26 10:53
5	1	("4994952").PN.	USPAT; US-PGPUB	2003/11/26 10:53
6	1	(("4994952").PN.) and high	USPAT; US-PGPUB	2003/11/26 11:21
7	278	link adj list with memory	USPAT; US-PGPUB	2003/11/26 11:22
8	72	(link adj list with memory) with pointer	USPAT; US-PGPUB	2003/11/26 11:23
9	61	(link adj list with memory) with pointer	USPAT	2003/11/26 11:23
10	1	((link adj list with memory) with pointer) same shared adj memory	USPAT	2003/11/26 11:23

US-PAT-NO: 5893162

DOCUMENT-IDENTIFIER: US 5893162 A

TITLE: Method and apparatus for allocation and management of shared memory with data in memory stored as multiple linked lists

---------- KWIC ---------

Abstract Text - ABTX (1):
Apparatus and methods for allocating shared memory utilizing linked lists are provided which are particularly useful in telecommunications applications such as ATM. A management RAM contained within a VLSI circuit is provided for controlling the flow of data into and out of a shared memory (data RAM), and stores information regarding a number of link lists and a free link list in the shared memory, and a block pointer to unused RAM locations. A head pointer, tail pointer, block counter and empty flag are stored for each data link list. The head and tail pointers each include a block pointer and a position counter. The block counter contains the number of blocks used in the particular queue. The empty flag indicates whether the queue is empty. The free link list includes a head pointer, a block counter, and an empty flag. Each memory page of the shared data RAM receiving the incoming data includes locations for storing data. The last location of the last page in a block of shared data RAM memory is preferably used to store a next-block pointer plus parity information. If there are no more blocks in the queue, that last location is set to all ones. An independent agent is utilized in the background to monitor the integrity of the link list structure. Using the methods and apparatus of the invention, four operations are defined for ATM cell management: cell write, cell read, queue clear, and link list monitoring.

Brief Summary Text - BSTX (13):
In accord with the objects of the invention a management RAM

contained within a VLSI is provided for controlling the flow of data into and out of a shared memory (data RAM). The management RAM is preferably structured as an x by y bit RAM which stores information regarding y-2 data link lists in the shared RAM, a free link list in the shared RAM, and a block pointer to unused shared RAM locations. Information stored in the x bits for each data link list includes a head pointer, a tail pointer, a block counter and an empty flag. In a preferred embodiment particularly applicable to the control of ATM data, the head and tail pointers are each composed of a block pointer and a position counter, with the position counter indicating a specific page in a block which is made up of a set of contiguous pages of memory, and the block pointer pointing to the block number. Regardless of how constituted, the head pointer contains the address of the first word of the first memory page of the link list, and the tail pointer preferably contains the address of the first word of the last memory page in the link list. The block counter contains the number of blocks used in the particular queue, and has a non-zero value if at least one page is used in the queue. The empty flag indicates whether the queue is empty such that the content of the link list should be ignored if the queue-empty flag indicates that the queue is empty.

Detailed Description Text - DETX (3):

The managing RAM 162 may serve various functions, including providing information for assisting in the processing of the header of the ATM cell as discussed in the parent application hereto. For purposes of this invention, however, the managing RAM 162, or at least a portion thereof, is preferably provided as a x bit by y word RAM for the purpose of managing y-2 link lists which are set up in the shared RAM 180 (y-2 equalling the product of w ports times v priorities). Thus, as seen in FIG. 2, a link list information structure for y-2 data queues includes: a head pointer, a tail pointer, a block counter, and a queue empty flag for each of the y-2 data queues; a free list block pointer, block counter, and queue empty flag for a free list; and a block

pointer for the unused blocks of memory. Each head pointer and tail pointer
preferably includes a block pointer and a position counter, with the block
pointer used for pointing to a block in the memory, and the position counter
being used to track pages within a block of memory. Thus, for example, where
ATM cells of fifty-three bytes of data are to be stored in the shared memory,
and each cell is to be stored on a "page", a block having four contiguous pages
may be arranged with the position counter being a two bit counter for
referencing the page of a block. The block counter for each queue is used to
reference the number of blocks contained within the queue. The queue empty
flag when set indicates that the queue is empty, and that the pointers
contained within the queue as well as the block count can be ignored.

Detailed Description Text - DETX (4):

As suggested above, the head pointer for each link list queue contains the
address of the first word of the first memory page of the queue in the shared
memory. The tail pointer for each link list queue contains the address of the
first word of the last memory page in the queue. Each memory page of the
shared memory is composed of M contiguous memory addresses. Depending on the
memory type, each address location can be of size B bits, with common sizes
being eight bits (byte), sixteen bits (word), thirty-two bits, or sixty-four
bits. In accord with the preferred embodiment of the invention, the address
locations are sixteen bits in length with the first M-1 locations in a page
containing the stored information. The M'th location of a last page in a block
is used to store a next block pointer which is set to the first location of the
next block plus an odd parity bit. Where the block is the last block in the
queue, the Math location of the last page in the last block is set to all ones.
Where the page is neither the last page of the block, nor the last block in the
queue, the M'th location of the page is not utilized. In the preferred
embodiment of the invention used with respect to ATM telecommunications data,
each page is thirty-two words in length (i.e., M=32), with each word being
sixteen bits. Thus, an ATM cell of fifty-three bytes can be stored on a single

page with room to spare. It should be appreciated, that in some applications, only the data payload portion of the ATM cell (i.e., forty-eight bytes), and not the overhead portion (five bytes) will be stored in the shared memory. In other applications, such as in switches where routing information is added, cells of more than fifty-three bytes may be stored. Regardless, with a thirty-two word page, system addressing is simplified.

Detailed Description Text - DETX (7):

Turning to FIG. 3c, specifics are seen of the management RAM which would be associated with managing the shared memory in the state of FIG. 3a. In particular, information for link list #1 is seen with a head pointer having a block pointer having a value equal to 512 and a position counter set at "00" to indicate a first page of memory in the block storing data. The tail pointer of the link list #1 information has a block pointer having a value equal to 122 and a position counter set to "11" to indicate that all pages of block 122 are being used. The block counter of the information for link list #1 is set to a value of three, and the queue empty flag is not set (i.e., equals zero). Information for link list #2 is seen with a head pointer having a block pointer having a value equal to 511 and a position counter set at "01" to indicate that the data first occurs at a second page of the block (i.e., the first page already having been read from the block). The tail pointer of the link list #2 information has a block pointer having a value equal to 123 and a position counter set at "10" which indicates that there is no data in the last page of the block. The block counter of the link list #2 information is also set to a value of three, and the queue empty flag is not set. The value of the head and tail pointers and block count for the information of link list #N are not indicated, as the queue empty flag of link list #N is set (equals one), thereby indicating that the pointers and block counter do not store valid data.

Likewise, while details of information for other link lists are not shown, the only data of interest would be that the queue empty flags related to all of those link lists would equal one to indicate that no valid data is

being stored
with reference to those link lists. The head pointer of the free list information has a block pointer set to a value 510, and a block count of 385.
The queue empty flag of the free list is not set, as the free list contains
data. Finally, the block pointer relating to the Unused queue is shown set to
a value of 121. It is noted that in order to increase performance, the free
list head pointer and block counter information is preferably implemented in a
series of flip-flops, and is thus readily available for purposes discussed
below with reference to FIGS. 4a-4d. The queue empty flags are also preferably
similarly implemented.

Detailed Description Text - DETX (13):
If at 256 it is determined that the cell which has been read out of shared
memory is the last in a block, then at 266, the head pointer for the free list
as obtained from the management RAM is inserted into the last word of the last
page of the freed block. Then, at 268, the queue status for the link list is
updated by changing the block pointer and position counter of the head pointer
(to the value contained in the last word of the page of memory being read out
of the shared memory), and by decrementing the block counter. Again, it is
noted that if the free list was empty prior to adding the freed block, the free
list must be initialized (with appropriate head pointer and block counter) and
the queue empty flag changed, and the last word in the freed block in the
shared RAM should be set to all ones. It is also noted, that upon obtaining
the pointer in the M'th location of the last page of the block, according to
the preferred embodiment of the invention, at 270, a parity check is done on
the pointer. At 272, the calculated parity value is compared to the parity bit
stored along with the pointer. Based on the comparison, at 274, a parity error
condition can be declared, and sent as an interrupt message via the
microprocessor interface port 167 (FIG. 1) to the microprocessor (not shown).
Preferably, when a parity error is found, the microprocessor treats the situation as a catastrophic error and reinitializes the management and data
RAMs.

Claims Text - CLTX (46):
said control means includes means for comparing a sum of counts of said block counters of each link list containing data, said free link list, and said unused pointer to the number of blocks in said shared memory means.

Claims Text - CLTX (83):
said control means includes means for comparing a sum of counts of said block counters of each link list containing data, said free link list, and said unused pointer to the number of blocks in said shared memory means, and means for generating an error signal is said sum of counts does not equal said number of blocks in said shared memory means.